Malicious URL Detection and Identification

Anjali B. Sayamber
PVPIT, Pune, India.

Arati M. Dixit
PVPIT, Pune, India.

ABSTRACT

Malicious links are used as a source by the distribution channels to broadcast malware all over the Web. These links become instrumental in giving partial or full system control to the attackers. This results in victim systems, which get easily infected and, attackers can utilize systems for various cyber crimes such as stealing credentials, spamming, phishing, denial-of-service and many more such attacks. To detect such crimes systems should be fast and precise with the ability to detect new malicious content. This paper introduces various aspects associated with the URL (Uniform Resource Locator) classification process which recognizes whether the target website is a malicious or benign. The standard datasets are used for training purpose from different sources. The rising problem spamming, phishing and malware, has generated a need for reliable framework solution which can classify and further identify the malicious URL. An alternative approach has been proposed which uses a Naïve Bayes classifier for an automated classification and detection of malicious URLs. The proposed model based on Naive Bayes is supported by clustering and classification technique. On the other hand, they are rarely used for general probabilistic learning and inference which is typically used for estimating with conditional and marginal distributions. The proposed work in this paper shows that, for a wide range of benchmark datasets, Naive Bayes models learned using Probability model has better accuracy than Support Vector Machine model.

General Terms

Algorithms, Attack Types.

Keywords

Machine Learning, Feature Extraction, Benign, Malicious Web Pages, Classification Module, Web-Based Attacks.

1. INTRODUCTION

As any file on a computer is to be found by giving its filename, similarly to trace any Web site its Uniform Resource Locators (URLs) are used. One can retrieve a site by typing a URL into the address bar of browser or simply by clicking correct URL one can access desired website. E.g. https://mail.google.com/mail/#inboxIt follows standard syntax :< protocol>< hostname><path>. Malicious Web sites covers a range of different illicit enterprises which are unsafe to visit, that’s why different types of malicious sites allocate various threats to users. If type of this threat is known it will be easy to inspect these types independently and understand their features which will be helpful to track the malicious site and to find out solution against a particular kind of threat. Three major categories of malicious sites (Spamming, Phishing, Malware) are considered in this paper, and each class is separated from the other by level of interaction required by the user.

A simple probabilistic classifier based on applying Bayes theorem from Bayesian statistics with strong naïve independence assumptions is known as Naive Bayes classifier [24]. In more detail the fundamental probability model is described as “independent feature model" [24]. In simple terms, a Naive Bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 2.5" in diameter. Each property has its independent contribution to the probability that this fruit is an apple, Even if these features depend on each other or upon the existence of the other features.

For supervised learning Process Naive Bayes classifiers can be used for training it works very efficiently, with the help of precise characteristics of the probability model. Naïve Bayes Classifier technique is mostly preferred when the dimensionality of the inputs is high. In spite of simplicity of Naive Bayes, it can handle and perform better than more complicated classification methods. Naïve Bayes model can be used for identifying the patients having heart disease by determining characteristics of patients. It calculates the probability of each input attribute independently for the expected state. Maximum likelihood method is used by many real time applications for parameter estimation, it can work without making an allowance for or using any Bayesian methods. In 2004, work on analysis of the Bayesian classification problem demonstrates that it shows outstanding performance by giving some theoretical causes for effectiveness of Naive Bayes classifiers [1]. After two year i.e. in 2006 it is analyzed that Bayes classification is outperformed by supplementary approaches for e.g. boosted trees or random forests [2]. After broad comparison it is concluded that small amount of data is enough for training purpose. For classification purpose it is mandatory to calculate means and variances of the variables. As independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The Figure 1 shows information related to URL and URL Classification. It reflects - type of URL, features [19], [20], [21], [22], datasets [15], [16], [17], [18], learning approaches, models [1], [26], and attack types [23] related to URLs. These URLs exhibit various features like: Lexical, Link Popularity, DNS, DNS Fluxiness, Network, and webpage content .Type of URL can be typically seen as benign and malicious URLs. Malicious URLs also further classified on the basis of attack types such as:

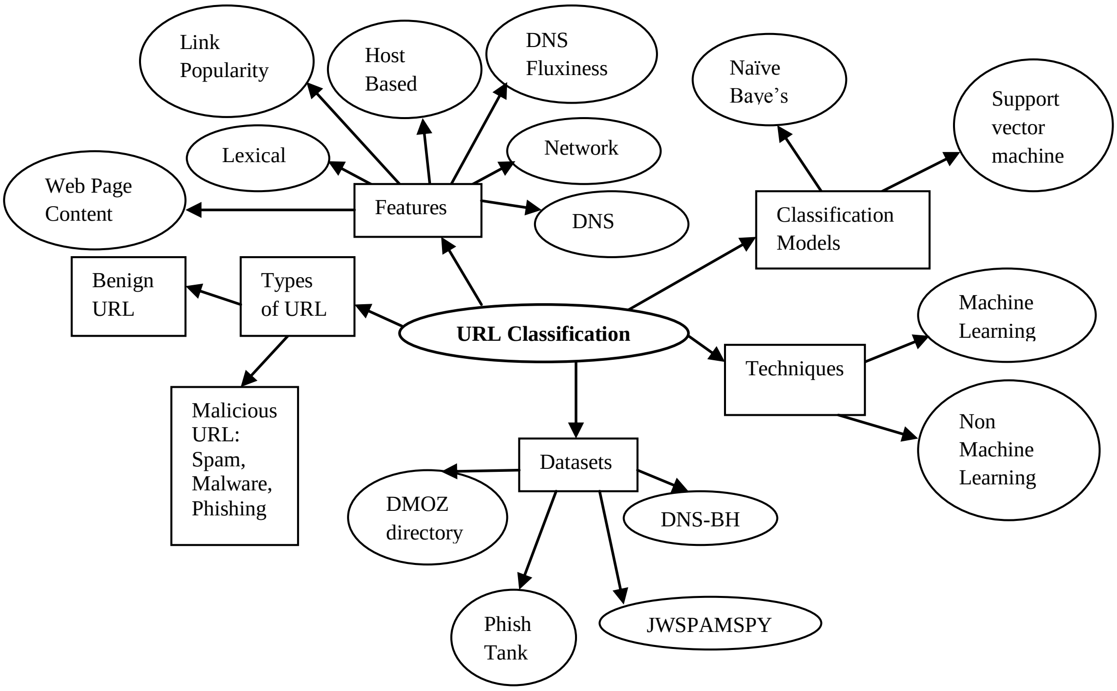


Figure: 1 URL Classification [23]

Spamming, phishing, and malware. Beginning with an overview of the classification problem, for which trained datasets are used as a collection of URLs, followed by a discussion of the learning approaches used for classification on basis of features, and finally SVM and Naïve Bayes are classifier used for the URL classification.

A brief review of literature to understand the background and related work is presented in section 2. The proposed Malicious URL Detection and Identification model using Naïve Bayes for general probability estimation is discussed in Section 3. The observation and experimental analysis of proposed model are articulated in section 4. The section 5 consists of concluding remarks associated with the proposed model.

2. RELATED WORK

As web Attacks are increasing rapidly, it is essential to find out cause of such attacks.URL classification has been topic of interest for many researchers in the area of privacy and security. Phishing is an issue related to the false E-mail in which innocent users get trapped by malicious web sites, these sites get access to the private information of user. A content based approach CANTINA, detects phishing web links [27]. Web spam is a problem associated with falsely created pages into the web sites, when user click on spam URL it redirects user to the harmful pages which seems to be very real but it drive traffic to the pages which are used for fun or profit or to mal-advertize the people. To inspect such spam pages or to identify these malicious sites various techniques are used together with the help of classification algorithms [28]. Malware [23]: It is short form for malicious software*; it can be in the form of code, scripts, active content, and other software.* Various [25] online learning methods for detecting malicious Web links and for identification purpose uses lexicon and host-based features of the related URLs. It observes that use of online algorithm is appropriate because the distribution of features alters constantly, which characterizes malicious URLs. The support vector machine (SVM) is machine learning algorithm used for binary classification problems. SVM is based on the concept where input vectors are non-linearly mapped to a very high dimension feature space. In this feature space a linear decision surface is constructed by creating functional margin between two classes. The idea behind the support vector network was previously implemented for the restricted case where the training data can be separated without errors. This paper [26] extends previous result on nonlinear training data. In recent days the Bayesian networks are considered to be efficiently representing multifaceted probability distributions, and therefore have created interest amongst the researchers [11]. But, this efficiency does not extend to inference, which is usually #P-complete [12]. The exact inference methods in Bayesian networks in practice have been experienced to be costly, and thus alternative options for approximate methods like Markov chain Monte Carlo [9] and loopy belief propagation [13] are considered. Naïve Bayes classification model is extensively based on Bayesian network [7] and clustering [5].Advantage of Naive Bayes is that it has combination of many components, but all variables within each component are assumed to be self-determining with respect to each other, so it can work on small size data for its training. Probability estimation algorithm is used in case of few unnoticed variables, which is used for computing unobserved values using the current parameters and computing the maximum likelihood or Mapping parameters are used for the current expectations [6]. When the structure is unknown, it can be learned by using some user defined values or starting with an empty or previous network and greedily adding, deleting and reversing arcs to optimize some score function [4] [10], which compares predictive accuracy of various components. It is based on set of assumptions from previous knowledge and statistical data which is used for learning Bayesian classifier. Learning structure given incomplete data requires a computationally expensive combination of expectation maximization (EM) and structure search. It shows how to apply EM and structural expectation maximization (SEM) to CTBNs [8].

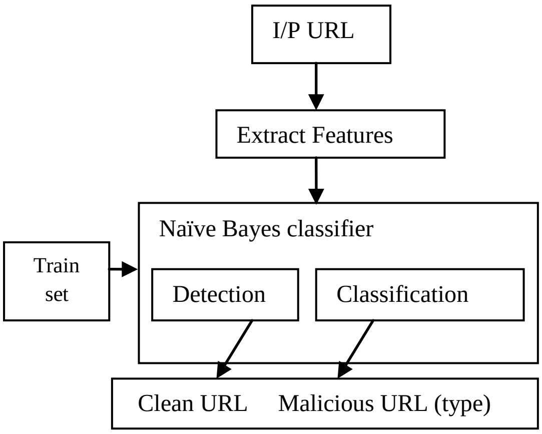


Figure 2: The framework of proposed method.

When Naive Bayes model is used for learning from train dataset, it never contains more attributes than components available and it guarantee that assumption will be brought up effectively. Similar experiments were evaluated on URLs using Support Vector Machine [14] instead of Naïve Bayes. The proposed work in this paper represents experimental evaluation for classification and detection of URLs using Naïve Bayes and compares the results of SVM and NB models. The Support Vector Machines (SVM) is a classification algorithm which is used for creating functional margin that helps to discover best hyper plane between two classes of data, by separating positive and negative examples through solid line in the middle called decision line [14], whereas Naive Bayes classifier is basically a probabilistic classifier based on assumption. On the basis of assumption and learning from train set; it finds out most suitable assumption based on previous assumptions and initial knowledge.

3. PROPOSED WORK

3.1 Overview

Proposed work is done for identifying malicious web links and identifying the type of attack by using Naïve Bayes Algorithm. An attempt on the similar lines [20] uses SVM, ML-KNN, and RAKEL algorithm. This work is done for comparing the results of both algorithms. Proposed framework works in three stages as shown in Figure 2:

- Stage 1: consist of training data collection,
- Stage 2: supervised learning with the training data,
- Stage 3: malicious URL detection and attack type Identification.

These stages can operate consecutively as in batched learning, or in an interleaving manner: additional data is collected to incrementally train the classification model while the model is used in detection and identification.

3.2 Proposed Modules

Proposed work is executed in three modules which are explained below.

3.2.1 Training Data

Stage 1 consists of training data collection as shown in Figure 3. Training Data is URLs of known type means its Domain name and category is given in train set. Example: 1) Pempoo.com phishing

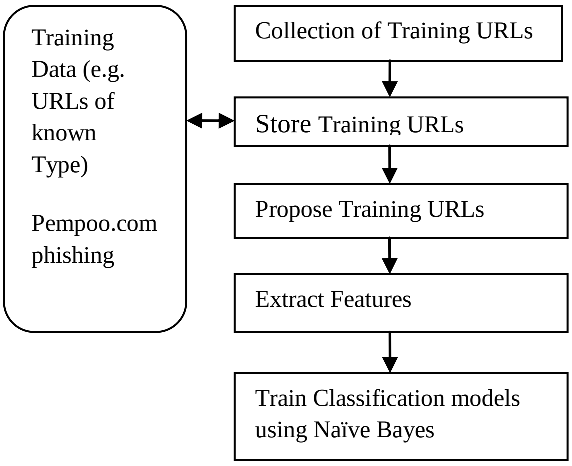

Training Data (e.g. URLs of known Type)
Pempoo.com phishing
Collection of Training URLs
Store Training URLs
Propose Training URLs
Extract Features
Train Classification models using Naïve Bayes

Figure 3: Training Data

In this e.g. Pempo.com is Domain name and phishing is a category of known URL. Numbers of URLs are stored as train set with their known category used for learning process. This Train set is submitted for feature extraction, and thus the learning process is based on Feature Extractions.

3.2.2 Feature Extraction Module

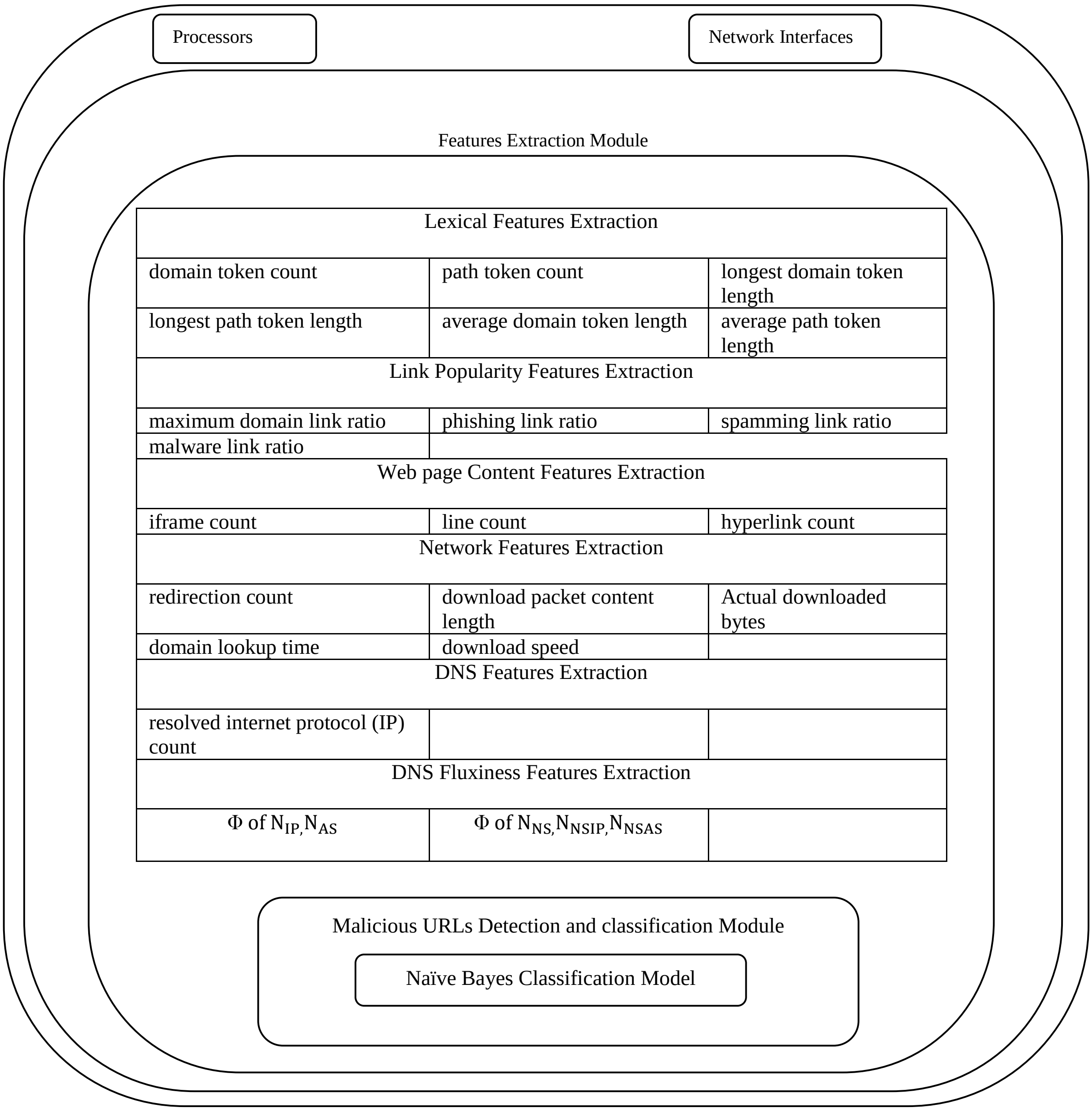
Feature Extraction Module is based on six features: Lexical Feature, Link Popularity Feature, Web page Content Feature, Network Feature, DNS Feature, and DNS Fluxiness Feature shown in Figure 4.

A. Lexical Features

Malicious URLs, esp. those for phishing attacks, usually have distinguishable patterns in their URL. Among these lexical options, the typical domain/path token length (delimited by '.', '/', '?', '=', '-', '') and name presence were driven from a study by McGrath and Gupta [19] that phishing URLs show completely different lexical patterns.

B. Link Popularity Features

One of the foremost necessary options utilized in this technique is "link popularity", that is calculable by examination of the amount of incoming links from alternative websites. Malicious sites tend to possess a less amount of link popularity, whereas several benign sites tend to possess a highest amount of link quality. Each link popularity of an address and link popularity of the URL's domain are utilized in this technique.


Processors
Network Interfaces
Features Extraction Module
Malicious URLs Detection and classification Module
Naïve Bayes Classification Model

Lexical Features Extraction		
domain token count	path token count	longest domain token length
longest path token length	average domain token length	average path token length
Link Popularity Features Extraction		
maximum domain link ratio	phishing link ratio	spamming link ratio
malware link ratio		
Web page Content Features Extraction		
iframe count	line count	hyperlink count
Network Features Extraction		
redirection count	download packet content length	Actual downloaded bytes
domain lookup time	download speed	
DNS Features Extraction		
resolved internet protocol (IP) count		
DNS Fluxiness Features Extraction		
Φ of $N_{IP,}N_{AS}$	Φ of $N_{NS,}N_{NSIP,}N_{NSAS}$	

Figure 4: Feature Extraction Module

C. Webpage Content Features

Recent development of the dynamic webpage technology has been exploited by hackers to inject malicious code in to sites through commerce and so activity exploits in webpage content. Therefore, applied math properties of client-side code within the online page are used as options to observe malicious sites. To extract webpage content options (CONTs), users have a tendency to count the numbers of HTML tags, iframe, lines, and hyperlinks within the webpage content [20].

D. DNS Features

The DNS options are a unit associated with the name of an address. It is found that most spam is being sent from a few regions of IP address space, and that spammers appear to be using transient "bots" that send only a few pieces of email over very short periods of time [21]. This shows that a major portion of spammers came from a comparatively little assortment of autonomous systems.

E. DNS Fluxiness Features

A freshly rising fast-flux service network (FFSN) establishes a proxy network to host extralegal online services with a really high convenience. To detect URLs which are served by FFSNs, it uses the discriminative features proposed by Holz et al. [22]

$$\varphi = \frac{N_{IP}}{N_{Single}}$$

Where,

φ = Fluxiness,N_{IP} = Total number of unique IPs,
N_{Single} = number of IPs (a single lookup returns.)

F. Network Features

Attackers could attempt to hide their websites exploitation with the help of multiple redirections such as iframe redirection and address shortening. Benign sites and malicious sites have different values for redirection count.

3.2.3 Classification Module

In Classification Module as shown in Figure 5 an unknown URL, whose Domain name is given and class i.e. Type of URL is unknown which user want to identify. Unknown URL given for testing is submitted to Extract Features associated with URL, and maps these features with extracted features from known train set. Mapping is based on Classification Model (Naïve Bayes) is applied to detect a Malicious URL and classify the Malicious URL.

3.3 DATA SETS

Benign URLs were collected from two sources 1) DMOZ Open Directory, 2) Yahoo!'s directory [18]. Malicious URLs were collected from the following sources: The spam URLs were acquired from jwSpamSpy [16] which is known as an e-mail spam. The phishing URLs were acquired from Phish Tank [17], it is a free community site where anyone can submit, verify, track and share phishing data. The malware URLs were obtained from DNS-BH [15]. Similar dataset references are used for training purposes in SVM based model [20].

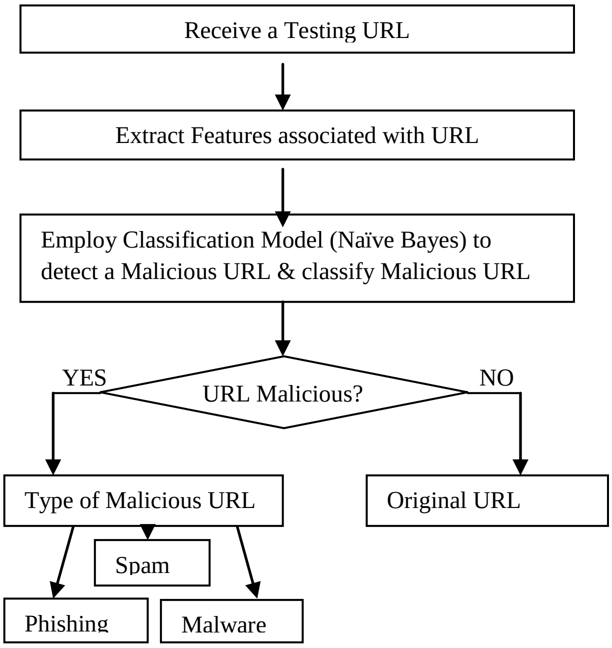

Receive a Testing URL
Extract Features associated with URL
Employ Classification Model (Naïve Bayes) to detect a Malicious URL & classify Malicious URL
YES
URL Malicious?
NO
Type of Malicious URL
Original URL
Spam
Phishing
Malware

Figure 5: Classification Module

3.4 Mathematical Model

Naive Bayes Rule is the basis for many machine-learning and data mining methods. The rule (algorithm) [24] is used to create models with predictive capabilities. It provides new ways of exploring and understanding data. It is used when data is high and we want efficient output compared to other methods.

The probability model for a classifier is a conditional model over a dependent class variable C.

$$p(C|F_{1,......,}F_n)$$

Using Bayes' theorem,

$$p\big(C|F_{1,......,}F_n\big) = \frac{p(C)\;p(F_{1,......,}F_n/C)}{p(F_{1,......,}F_n)}$$

•p(C | $F_{1,......,}F_n$) = probability of instance $F_{1,......,}F_n$ being in class C.

•p($F_{1,......,}F_n$ | C) = probability of generating instance $F_{1,......,}F_n$ by given class C, One can imagine that being in class C, causes to have feature $F_{1,......,}F_n$ with some probability.

•*p(C)* = probability of occurrence of class C,

•*p*($F_{1,......,}F_n$*)* = probability of instance $F_{1,......,}F_n$ occurring.

In simple words the above equation can be written as

$$\text{Posterior} = \frac{prior\ *\ likelihood}{evidence}$$

The denominator is independent of C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p\big(C, F_{1,......,}F_n\big)$$

Naïve Bayes is an classification approach mostly used for detection and categorization of text documents. By providing a set of classified training samples, an application can learn from these examples, so as to predict the class of unknown URL. With a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. The features (*F1, F2, F3, F4*) which are present in URL are independent from each other. Every feature Fi(1<=i<=4) text binary value showing whether the particular property comes in URL. The probability is calculated that the given web belongs to a class m (m1: Non-phishing and m2: Phishing) as follows:

*P (m1/F) = (P (m1)*P (F/mi))/P (F)*

Where all of P (F) are constant meanwhile *P (Fi|m1)* and *P(mi)* can be easily calculated from training. The proportional to *P (m1|F), P(m2|F)* is calculated and the results are as follows:

P(m1|F)P(m2|F) > b (b>1), Benign link.

P(m2|F)P(m1|F) > b , Malicious link.

3.5 The Proposed Algorithm based on Naive Bayes

INPUT: Training set, URLs to be tested.
OUTPUT: testing domain names with their attack type.

Step 1: For given feature calculate its sub features for training purpose using the training set.
Step 2: The classifier is created from the training set using a Gaussian distribution and by calculating mean and variance of each sub feature.
Step 3 : Probability of individual class is calculated.
Step 4: Testing sample with their calculated feature is taken for classification.
Step 5: Posterior for each class (Benign, Spam, Phishing, and Malware) is calculated.
Step 6: Analyze posterior values of each class.
Step 7: Among Four classes, class with greater value of posterior is assigned to testing domain.

4. EXPERIMENTAL ANALYSIS

For experimental purpose different datasets are used having variation in size. As Naïve Bayes can independently estimate dimensions of distribution, so it does not depend on size of Train set, and results on Test set vary randomly various size of Test set. The comparison of performance of SVM[14] based and Naïve Bayes based classification model is shown in Figure 6. The y-axis shows the percentage of accurate URL classification and detection with x-axis showing results with respect to various features like lexical, web, DNS, DNSF, etc. Figure 7 reflects: The y-axis shows the percentage of accurate URL classification and detection with x-axis showing results with respect to various URL Types like Benign, Spam, Phishing, Malware. The experimental result shows that Naïve Bayes based classifier exhibits higher accuracy than Support Vector Machine algorithm for almost all features and URL Type.

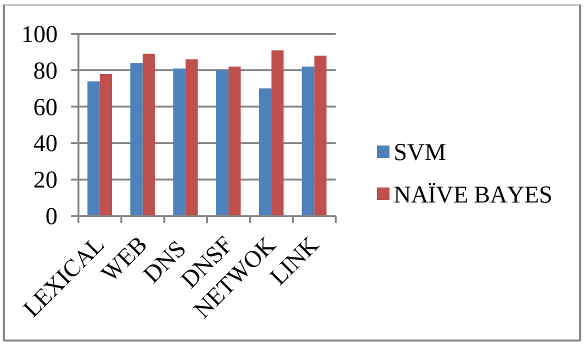


Figure 6: Comparison Graph Based on Features

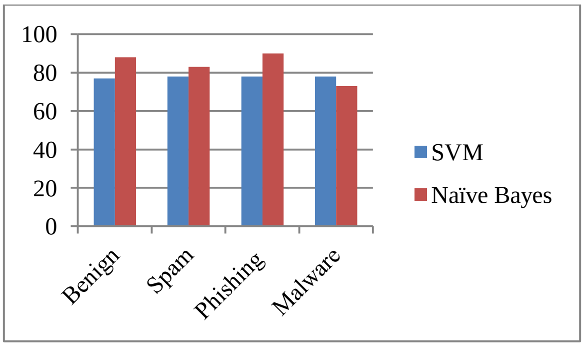


Figure 7: Comparison Graph Based on Attack Types

5. CONCLUSION

Malicious links are well-known weapons used by attackers to acquire control of victim systems, which can be utilized to execute cyber crimes involving spamming, phishing, denial-of-service and many more. The rising levels of cyber crimes have necessitated the requirement of reliable classification and identification framework. To detect and prevent such crimes a URL classification and identification model is proposed based on Naïve Bayes classifier. It is observed that the proposed Naive Bayes model is more accurate than support vector machine for detection and identification of type of malicious URLs with the help of probability estimation tasks. Experiments on a large number of datasets show that these two algorithms take almost same time to learn, but Naive Bayes reasoning is relatively faster. The Naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to improve problems resulting from the annoyance of dimensionality, like the need for data sets that scale exponentially with the number of features. To the best of our knowledge the proposed work for a wide range of benchmark datasets, Naive Bayes models learned using Probability model has better accuracy than Support Vector Machine model.

6. REFERENCES

[1] Harry Zhang "The Optimality of Naive Bayes". FLAIRS 2004 conference.

[2] Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning, 2006.

[3] George H. John and Pat Langley “Estimating Continuous Distributions in Bayesian Classifiers”. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo, 1995.

[4] Breese, J. S., Heckerman, D., & Kadie, C. “Empirical analysis of predictive algorithms for collaborative filtering”. *Proc. UAI-98,* (1998), (pp. 43–52).

[5] Cheese man, P., & Stutz, J. (1996). Bayesian classification (Auto Class): Theory and results. In *Advances in knowledge discovery and data mining*, 153–180. Menlo Park, CA: AAAI Press.

[6] Dempster, A. P., Laird, N.M., & Rubin, D. B. (1977). MaCmum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society B*, *39*, 1–38.

[7] Domingo’s, P., & Pazzani M.. “On the optimality of the simple Bayesian classifier under zero-one loss”. *Machine Learning*, *29*, 103–130, (1997)..

[8] Friedman, N. (1998). The Bayesian structural EM algorithm. *Proc. UAI-98* (pp. 129–138),

[9] Gilks, W. R., Richardson, S., & Spiegel halter, D. J. (Eds.).(1996). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall.

[10] Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statist. data. *Machine Learning*, *20*, 197–243.

[11] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufmann.

[12] Roth, D. (1996). On the hardness of approCmate reasoning. *Artificial Intelligence*, *82*, 273–302.

[13] Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2001). Generalized belief propagation. In *Adv. NIPS 13*, 689–695.

[14] Hyunsang Choi.. Seoul, Bin B. Zhu.**“**Detecting Malicious Web Links and Identifying Their Attack Types”. Korea University (2011).

[15] DNS-BH. Malware prevention through domain blocking.

[16] JWSPAMSPY. E-mail spam filter for Microsoft Windows.

[17] PHISHTANK. Free community site for anti-phishing service.

[18] http://random.yahoo.com/bin/ryl)3. (accessed on 20/06/2014)

[19] Mcgraph, D. K., And Gupta, M. (2008). Behind phishing: An examination of phisher modi operandi. In LEET: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats.

[20] Hou, Y.-T., Chang, Y., Chen, T., Laih, C.-S., And Chen, C.-M. "Malicious web content detection by machine learning". Expert Systems with Applications (2010), 55–60.

[21] Ramchandran, A., And Feamster, N. "Understanding the network-level behavior of spammers". In Sigcomm (2006).

[22] Holz, T., Gorecki, C., Rieck, K., And Freiling, F. C. "Detection and mitigation of fast-flux service networks". In NDSS: Proceedings of the Network and Distributed System Security Symposium (2008).

[23] Anjali B. Sayamber, Arati M. Dixit. "On URL Classification" International Journal of Computer Trends and Technology (IJCTT) – volume 12 number 5 – Jun 2014.

[24] http://en.wikipedia.org/wiki/Malware (accessed on 20/06/2014).

[25] Fette, I., Sadeh, N., and Tomasic, A. "Learning to detect phishing emails". In WWW: Proceedings of the international conference on World Wide Web (2007).

[26] Cortes, C., and Vapnik, V. "Support vector networks". Machine Learning (1995), 273–297.

[27] Zhang, Y., Hong, J., and Cranor, L. Cantina: "A content-based approach to detecting phishing web sites". In WWW: Proceedings of the international conference on World Wide Web (2007).

[28] Ntoula, A., Najork, M., Manasse, M., and Fetterly,D. "Detecting spam web pages through content analysis". In WWW: Proceedings of international conference on World Wide Web (2006).